DC パワーサプライ

PSH/PSS/PST シリーズ

プログラムマニュアル

GW INSTEK PART NO

ISO-9001 CERTIFIED MANUFACTURER

GW INSTEK

このマニュアルは著作権によって保護された知的財産情報を含んでいます。 グッドウィルはすべての権利を保持します。 グッドウィルの文書による事前承諾なしでこのマニュアルを複写、再現、翻訳することはできません。

このマニュアルに記載された情報は印刷時点のものです。 グッドウィルは製品改良のため、いつでも予告なしで仕様、機器、および保守手順を変更する権利を保有します。

Good Will Instrument Co., Ltd.
No. 7-1, Jhongsing Rd., Tucheng City, Taipei County 236, Taiwan.

目次

1 はじめに 1

2 GP-IB インタフェース 2

3 RS-232C インタフェース 5

4 入力キューと出力キュー 8

5 コマンドと構文 8

6 詳細コマンドリファレンス 18

7 ステータスとエラー報告 36

1 はじめに

先進的な自動計測システムでは、計測器とコンピュータとの間の通信が不可欠です。測定手法をユーザーの試験プログラムに応じて変更することができます。このため、プログラマブル電源を計測コントローラあるいはコンピュータから RS232 インタフェース(オプション)または GP-IB(オプション)経由で制御することができます。

インタフェースの選択と設定

GP-IB アドレスは通常動作状態で変更することができます。前面パネルの[SHIFT]キーと[LOCAL]キーを押すと、最後に使用されていた通信インタフェース設定が表示されます。

インタフェースを選択して[ENTER]を押した後、ボーレート(または GP-IB アドレス)選択して[ENTER] を押してジョグダイアルによる設定を確認します。最後に「save」を選択して[ENTER]を押し、設定を保存します。

2 GP-IB インタフェース

GP-IB インタフェースの機能

プログラマブル電源の GP-IB インタフェースは IEEE-488.1-1987, IEEE-488.2-1992 および SCPI-1994 の規格に対応します。GP-IB インタフェースの機能を以下に示します。

SH1(ソースハンドシエーク)	プログラマブル電源は GP-IB にマルチレーンメッセージを送信できます。
AH1(アクセプタハンドシエーク)	プログラマブル電源は GP-IB からマルチレーンメッセージを受信できます。
T6(トーカー)	トーカーインタフェース機能には、基本トーカー、シリアルポール、およびアンアドレスイフ MLA 機能が含まれます。トークオンリーモードの機能は含まれていません。
L4 (リスナー)	コントローラが ATN (アテンション) ラインをアサートしてリスンアドレスを送信したときには、プログラマブル電源はリスナーになります。プログラマブル電源はリスンオンリー機能を持っていません。
SR1 (サービスリクエスト)	プログラマブル電源は、サービスを必要とするときには SRQ (サービスリクエスト) ラインをアサートしてコントローラに知らせます。
RL1 (リモート/ローカル)	プログラマブル電源は GTL(ゴートゥローカル) と LLO(ローカルロックアウト) のインタフェースメッセージの両方に応答します。
注意	PST と PSS は、OUTPUT オンの状態でローカルロックアウトすると OUTPUT はオンのままです。PSM は、OUTPUT オンの状態でローカルロックアウトすると OUTPUT はオフになります。

PP0 （パラレルポール）	プログラマブル電源はパラレルポール機能を持っていません。
DC1 （デバイスクリア）	プログラマブル電源はデバイスを電源投入直後の状態に戻すデバイスクリア機能を持っています。
DT0 （デバイストリガ）	プログラマブル電源はデバイストリガのインタフェース機能を持っていません。
C0 （コントローラ）	プログラマブル電源は他のデバイスを制御することはできません。

GP-IB 接続に関する注記

プログラマブル電源を GP-IB システムに接続するときには、以下のことに注意してください。

- 1つの GP-IB バスに接続できるデバイスの数は最大 15 です。
- デバイスとの接続ケーブルは、20m 以内にしてください。
- 使用するケーブル 2m ごとに 1 つのデバイスを接続してください。
- バス上の各デバイスは独自のデバイスを必要とします。
  1つのデバイスを 2 つのデバイスが共用することはできません。
- GP-IB システムを使用するときにはシステムのデバイスの少なくとも 2/3 の電源をオンにしてください。
- GP-IB システムの接続形態をループあるいはパラレルにしないください。

パソコンの接続

プログラマブル電源を GP-IB インタフェース経由で制御するためには GP-IB カード付きのパソコンが必順です。

プログラマブル電源とパソコンの間を次のように接続します。

1. GP-IB ケーブルの片方の端をパソコンに接続します。
2. GP-IB ケーブルの他方の端をプログラマブル電源の GP-IB ポートに接続します。

3. プログラマブル電源の電源をオンにします。

4. パソコンの電源をオンにします。

GP-IB 接続の試験

GP-IB 接続が動作しているかどうかを調べたいときには、パソコンから GP-IB コマンドを送って調べることができます。例えば、クエリコマンド

*idn?

を送ると製造業者、型番、シリアル番号、およびファームウェアバージョンを以下のフォーマットで返すはずです。

WK.TMPRO,PST-3202,A000000,FW1.00

プログラマブル電源から適切な応答が得られない場合には、電源がオンであるか、GP-IB アドレスが適切か、ケーブル接続先がすべてアクティブかを調べてください。

3 RS-232C インタフェース

RS-232C インタフェースの機能

RS-232 インタフェースはパンコンとプログラマブル電源などの 2 台の機器間をポイントトゥポイントで接続するインタフェースです。両方の側でいくつかのパラメータを設定する必要があります。パラメータを設定した後、RS232 インタフェースを通してプログラマブル電源を制御することができます。

- ボーレート: 1200, 2400, 4800、または 9600 ボーを選択可能
- パリティビット: なし
- データビット: 8 ビット
- ストップビット: 1 ビット
- データフロー制御: なし

RS-232 接続に関する注意

プログラマブル電源は後面パネルに 9 ピン D 型 RS-232 コネクタを備えた DTE 機器です。9ピンコネクタ(オス型)のピン番号の割当てを図 2 に示します。プログラマブル電源を RS-232 インタフェースで接続したときには以下の点を確認してください。

- DTE デバイスの出力ラインを他 DTE デバイスの出力ラインに接続してはいけません。
- 多くの機器では 1 つあるいは複数の入力ピンを定常的に HI にしておく必要があります。
- 機器のシグナルグラウンドが外部機器のシグナルグラウンドと接続されていることを確認してください。
- 機器のシャーシグラウンドが外部機器のシャーシグラウンドと接続されていることを確認してください。
- 機器とパソコンとの接続ケーブルは 15m 以内にしてください。

- 機器で使用するボーレートとパソコン端末で使用するボーレートを合わせてください。
- ケーブル両端のコネクタと内部接続ラインが機器の要求に合致することを確認してください。

1. 接続なし
2. 受信データ (RxD)　(入力)
3. 送信データ(TxD)　(出力)
4. 接続なし
5. シグナルグラウンド (GND)
6. 接続なし
7. 接続なし
8. 接続なし
9. 接続なし

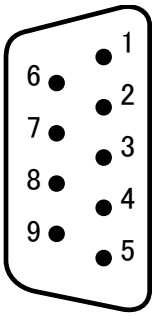

図 1 後面パネル DB-9-D RS232 コネクタのピン割当て

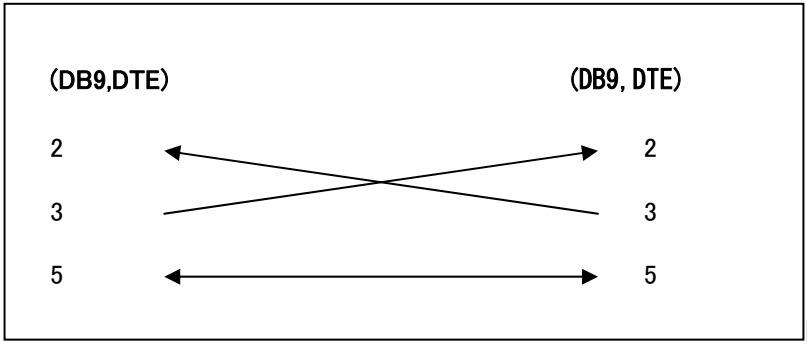

図 2 DB9 から DB9 への配線接続

パソコンとの接続

本器を RS-232 インタフェース経由で制御するためには COM ポート付きのパソコンが必要です。

本器とパソコンとの間の接続は以下のようにします。

1. RS-232 ケーブルの片方の端をパソコンに接続します。
2. ケーブルの他方の端をプログラマブル電源の RS-232 ポートに接続します。
3. プログラマブル電源の電源をオンにします。
4. パソコンの電源をオンにします。

RS-232 接続のテスト

RS-232 の接続が正常に動作しているかどうかを調べたいときには、パソコンからコマンドを送って調べることができます。
例えば、ターミナルプログラムを使ってクエリコマンド

*idn?

を送ると製造業者、型番、シリアル番号、およびファームウェアバージョンを以下のフォーマットで応答するはずです。

GW. INC,PST-3202,A000000,FW1.00

プログラマブル電源から適切な応答が得られない場合には、電源がオンであるか、RS-232 のボーレートが適切か、ケーブルの結線が正しいかを調べてください。

4 入力キューと出力キュー

128 バイトの入力キューと128バイトの出力キューが組み込まれており、保留中のコマンドとりターンメッセを保存して、リモートコントロールのための送信コマンドとりターンメッセージが失われないようにします。エラー/イベントキューの最大容量が20メッセージ分であるため、これらのバッファを使うことにより容量を超える入力データがあればデータが失われることに注意願います。

5 コマンドと構文

プログラマブル電源の GP-IB コマンドは IEEE-488.2 と SCPI 規格に適合しています。

SCPI

SCPI (プログラマブル計測器標準コマンド)は試験計測器の主要メーカーの国際団体が作成した規格です。 IEEE-488.2 の構文が SCPI で採用されており各種のプログラマブル計測器で同じ機能について同一のコマンドを利用できるようにします。

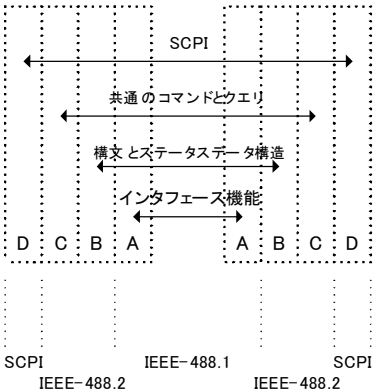

図 3 IEEE-488.1、IEEE-488.2 および SCPI の関係

図 3 に示すように、IEEE-488.1 規格はレイヤ A に位置します。

レイヤ A は GP-IB バスのインタフェース機能のプロトコルです。ソースハンドシエーク(SH)、アクセプタハンドシエーク(AH)、およびトーカーがこのレイヤに含まれています(合計 10 のインタフェース機能)。

レイヤ B では、構文とデータ構造が IEEE-488.2 規格全体の核心となることがあります。構文ではメッセージ通信の機能が定義されており、<プログラムメージ>(すなわち、「コマンド」)と<応答メッセージ>が含まれます。2 種類の<メッセージ>(すなわち、「コマンド」)と<応答メッセージ>が含まれます。2 種類のメッセージは、デバイスのコマンドと戻り値の構文構造を表します。データ構造は IEEE-488.2 規格によって定義される標準レポーテイングの構成です。

共通のコマンドとクエリがレイヤ C に含まれています。コマンドとクエリは、必須と任意の 2 つの種類に区分することができます。コマンドは制御設定の変更あるいは特定機能の実行を計測器へ命じます。クエリによって計測器がデータまたはステータス情報をパソコンに送り返します。コマンド末尾に疑問符(?)を付けると、クエリであることを示します。

注意 クエリコマンドで疑問符の前にスペース(空白;" _ ")があるコマンドと無いコマンドがあります。
例::CHANnel<x>:VOLTage_?
*idn?

レイヤ D は機器の情報に関連します。別々の機器は別々の機能を持ちます。SCPI コマンドセットはこのレイヤに属します。

コマンドの構文

SCPI に従って計測器に命令を伝えたいときには、以下の 3 つの基本要素を含める必要があります。

- コマンドヘッダ
- パラメータ(必要な場合)
- メッセージターミネータまたはメッセージセパレータ

コマンドヘッダ

コマンドヘッダは階層構造を持ち、コマンドツリー（図 4）によって表されます。

ツリーの再上位はルートレベルです。ルートノードがルートレベルに存在します。ルートノードと 1 つまたは複数の下位ノードがリーフノードと呼ばれる最終ノードへのヘッダパスを構成します。

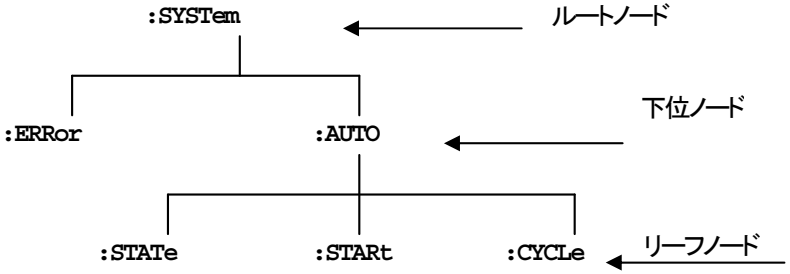

図 4: ツリー階層

コマンドヘッダはヘッダパスとリーフノードから構成されます。図 4 に示されたリーフノードのコマンドヘッダを図 5 に示します。

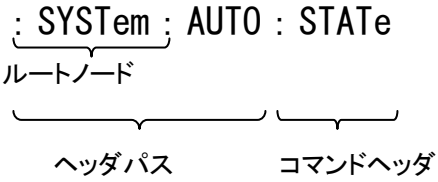

図 5 コマンドヘッダ

パラメータ

コマンドがパラメータを持つときには、その値を含める必要があります。コマニュアルではコマンドの構文を説明するときにパラメータの種類を示すために＜＞シンボルを使います。例えば、図 6 のコマンドの構文はブーリアン型のパラメータを含みます。

注意 パラメータの実際の値を入力するときには＜　＞、あるいはシンボルを含めないでください。

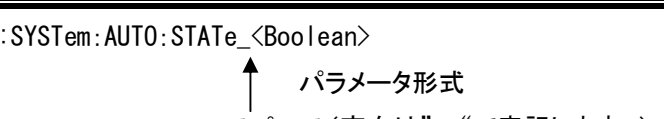

図 6 パラメータ付きのコマンドヘッダ

プログラマブル電源のブーリアン型および他の型のパランメータの定義を表 1 に示します。

パラメータの型	定義	例
<Boolean>	ブール代数値	0, 1
NR1	整数	0, 1, 18
NR2	小数	1.5, 3.141, 8.4
NR3	浮動小数点数	4.5E-1, 8.25E+1
ストリング	英数字	“No error”

表 1:構文記述のためのパラメータの型

メッセージターミネータおよびメッセージセパレータ

GP-IB メッセージターミネータ

IEEE 488.2 規格に従って、以下に示す任意のメッセージターミネータが利用可能です。

LF^END	改行コード(16 進 0A)および END メッセージ
LF	改行コード
<dab>^END	最終データバイトおよび END メッセージ

これらのターミネータはほとんどのアプリケーションプログラムと適合します。

1 つの行に複数のコマンドを記述するときにはセミコロンによってコマンドどうしを分離します。

RS-232 メッセージターミネータ

RS-232 バスではメッセージの終わりを示す信号がないため、LF （改行）をメッセージターミネータとして利用します。

一連のコマンドを計測器に送った後、メッセージターミネータの印として LF を付加する必要があります。クエリコマンドについては、計測器のリターンメッセージにも LF を付加してパソコンがメッセージターミネータを判別できるようにします。

コマンドの入力

プログラマブル電源のコマンドセットを規定する規格では、コマンド入力時にある程度の柔軟性を許容しています。例えば、多くのコマンドを短縮することができ、また複数のコマンドを組み合わせて 1 つのメッセージにしてプログラマブル電源に送ることができ。「フレンドリリスニング」と呼ばれる柔軟性によってプログラミング時間を節約してコマンドセットを覚えやすくし、また利用しやすくします。

コマンドの文字

プログラマブル電源はコマンドの大文字と小文字を区別しません。コマンドを大文字で入力しても小文字で入力してもかまいません。

スペース文字を含む任意のコマンドを実行することができます。しかし、パラメータとコマンドヘッダと間には少なくとも 1 個のスペースを入れなければなりません。

コマンドの短縮

ほとんどのコマンドには長形式と短形式があります。この節に示す各コマンドでは省略可能部分を小文字で示します。例えば、:CHANnel1:VOLTage_1.23 というクエリを: :CHAN1:VOLT_1.23 と省略することができます。

プログラマブル電源はコマンドがルートから始まっていると仮定するため、最初のコマンドヘッグをコロン（:）から始めることができます。

コマンドの結合

セミコロン(;)を使ってコマンドを結合することができます。
しかし、連続したクエリコマンドを使うとメッセージが失われるおそれがあります。

例: CHAN1:VOLT_?;CURR_?

セミコロンの後のコマンドがルートレベルとは別のヘッダパスを持つときには、コロンを使ってルートレベルに戻らなければなりません。

:CHAN1:VOLT_1.23;:OUTP:COUP:TRAC_1

セミコロンの後のコマンドが同一のヘッダパスを持つときには、コロンとパスを省略して新しいリーフノードのみを示すことができます。

例えば、

:CHAN1:VOLT_12.34;CHAN1:CURR_1.55

は次のコマンドと等価です。

:CHAN1:VOLT_12.34;CURR_1.55

複数のコマンドやクエリを結合して 1 つのメッセージにすることができます。

例えば、以下のようにすることができます。

:CHAN1:VOLT_12.34;VOLT_?

コマンドの概要

この節の表ではプログラマブル電源のコマンドの概要を示します。

表ではコマンドを以下の 3 種類に分類しています。

- 一般設定コマンド
- ステータスコマンド
- 各種コマンド

表では各コマンドの簡単な説明も含みます。

一般設定コマンド

プログラマブル電源の設定のコマンドとクエリを行う一般設定コマンドを表 2 に示します。

表 2: 一般設定コマンド

コマンド	説明
:CHANnel<x>:CURRent_<NR2>	★1 電流値を設定します。
:CHANnel<x>:CURRent_?	★1 電流値を返します。
:CHANnel<x>:VOLTage_<NR2>	★1 電圧値を設定します。
:CHANnel<x>:VOLTage_?	★1 電圧値を返します。
:CHANnel<x>:MEASure:CURRent_?	★1 実際の出力電流を返します。
:CHANnel<x>:MEASure:VOLTage_?	★1 実際の出力電圧を返します。
:CHANnel<x>:PROTection:CURRent <Boolean>	★1 過電流保護(OCP)をオンまたはオフに設定します。
:CHANnel<x>:PROTection:CURRent_?	★1 過電流保護(OCP)の状態(オンまたはオフ)
:CHANnel<x>:PROTection:VOLTage_ <NR2>	★1 過電圧保護(OVP)の値を設定します。
:CHANnel<x>:PROTection:VOLTage_?	★1 過電圧保護(OVP)の値を設定します。
:OUTPut:COUPle:TRACking_<NR1>	★2 プログラマブル電源が直列モード、並列モード、あるいは独立モードで出力するように設定します。
:OUTPut:COUPle:TRACking_?	★2 プログラマブル電源の動作モードを返します。
:OUTPut:PROTection:CLEar	過電圧、過電流、および過熱のエラーメッセージをクリアします。
:OUTPut:STATe_<Boolean>	出力状態(オンまたはオフ)を設定します。
:OUTPut:STATe_?	出力状態(オンまたはオフ)を返します。

注意:	“★1”の記号は、PSS シリーズおよび PSH シリーズでは<X>値が 1 のみであることを示します。
注意:	“★2”の記号は、PSS シリーズおよび PSH シリーズではこの機能を持たないことを示します。

ステータスコマンド

プログラマブル電源のステータスとイベント構造を構成する各種のレジスタとキューの設定と間合せを行うステータスコマンドを表 3 に示します。

表 3:ステータスコマンド

*CLS	ステータスデータ構造をクリアします。
*ESE_<NR1>	イベントステータスイネーブルレジスタ(ESER)を設定します。
*ESE?	イベントステータスイネーブルレジスタ(ESER)の内容を返します。
*ESR?	標準イベントステータスレジスタ (SESR)の内容を返してクリアします。
*SRE_<NR1>	サービスリクエストイネーブルレジスタ(SRER)の内容を返します。
*SRE?	サービスリクエストイネーブルレジスタ(SRER)の内容を返します。
*STB?	ステータスバイトレジスタ (SBR)を読み出します。
:STATus:OPERation:CONDition_?	オペレーションコンデイションレジスタの内容を返します。NRI を返します。
:STATus:OPERation:ENABle_<NR1>	オペレーションイベントレジスタのイネーブルマスタの内容を設定します。
:STATus:OPERation:ENABle_?	オペレーションイベントレジスタのイネーブルマスタの内容を返します。NRI を返しま

	す。
:STATus:OPERation:EVENt_?	オペレーションイベントレジスタの内容を問い合わせます。
:STATus:PRESet	オペレーションおよびクエスチョナブルステータスレジスタをプリセットします。
:STATus:QUEStionable:CONDition_?	オペレーションコンデイションレジスタの内容を返します。NRI を返します。
:STATus:QUEStionable:ENABle <NR1>	クエスチョナブルイネーブルレジスタのイネーブルマスタの内容を設定します。
:STATus:QUEStionable:ENABle_?	クエスチョナブルイネーブルレジスタの内容を問い合わせます。
:STATus:QUEStionable:EVENt_?	クエスチョナブルイベントレジスタの内容を問い合わせます。

各種コマンド

プログラマブル電源の管理機能を制御する各種コマンドを表 4 に示します。

表 4:各種コマンド

*IDN?	計測器の識別情報を返します。
*OPC	SESR のオペレーション完了ピットをセットすることによって、オペレーション完了時に過知します。
*OPC?	オペレーション完了時に通知します。 *OPC と同様ですが出力キューに1を返し SESR ピットのセットは行いません。
*RCL	★以前に保存したメモリから設定データを呼び出します。
*RST	保護レベルと状態をセットし、電流レベルと電圧レベルをゼロにリセットします。出力をオフにし、メモリアドレスをOOに設定します。
*SAV	★設定データをメモリに保存します。

*TST?	内部セルフテストを実施して結果を報告します。
*WAI	待って継続します。このコマンドによってコマンドを連続的に実行させます。このコマンドはIEEE-488.1-1987 によって必要とされます。しかし、プログラマブル電源はコマンドの連続的な実行が可能な設計になっています。
:SYSTem:AUTO:CYCLe_<NR1>	★実行回数を設定します。
:SYSTem:AUTO:CYCLe_?	★実行回数の設定を問合わせます。
:SYSTem:AUTO:DELay_<NR1>	★現在の応答メモリ状態での遅延時間を設定します。
:SYSTem:AUTO:DELay_?	★現在の応答メモリ状態で遅延時間を問合わせます。
:SYSTem:AUTO:END_<NR1>	★連続自動実行の最終メモリアドレスを設定します。
:SYSTem:AUTO:END_?	★連続自動実行の最終メモリアドレスを問い合わせます。
:SYSTem:AUTO:STARt_<NR1>	★連続自動実行の開始アドレスを設定します。
:SYSTem:AUTO:STARt_?	★連続自動実行の開始アドレスを問い合わせます。
:SYSTem:AUTO:STATe_<Boolean>	★自動実行のオン／オフを設定します。
:SYSTem:AUTO:STATe_?	★自動実行モードのオン/オフを返します。
:SYSTem:ERRor_?	エラー/イベントキューから次の項目を読み出します。
:SYSTem:MEMory?	★最終メモリアドレスを問い合わせます。
:SYSTem:VERSion?	SCPI のバージョンレベルを返します。

注意: “★”の記号は PSS および PSH がその機能を持たないことを示します。

6 詳細コマンドリファレンス

この章では各コマンドの詳細な説明を行います。各コマンドの例を示すとともにクエリと応答の例を示します。

*CLS （クエリなし）

機能:

イベントステータスデータレジスをすべてクリアします。出力キュー、オペレーションイベントステータレジスタ、クエスチョナブルイベントステータスレジスタ、および標準イベントステータスレジスタを含みます。

構文:

*CLS

例:

*CLS は、イベントレジスタをすべてクリアします。

*ESE

機能:

イベントステータスイネーブルウレジスタ(ESER)のピットの設定または間合わせを行います。ESER は標準イベントステータスレジスタ(SESR)をイネーブルしてステータスバイトレジスタ(SBR)のピット 5 (ESB)に要約します。

構文:

*ESE_<NR1>

*ESE?

<NR1>は、0～255.

戻り値:

<NR1> は 0～255 の数値であり、ESER の 2 進ピットの 10 進値を示します。

例:

*ESE 65 を送信することで、ESER を 2 進(01000001)に設定します。

ESER が 2 進の 100000010 を含む場合には、*ESE?によって 130 という値が返されます。

*ESR? (クエリのみ)

機能:

標準イベントステータスレジスタ (SESR)の内容を返してクリアします。

構文:

*ESR?

戻り値:

<NR1>は 0～255 の数値であり ESER のバイナリビットの 10 進値を示します。

例:

ESER が 2 進 11000110 を含む場合には、 *ESR?によって 198 という値が返されます。

*IDN? (クエリのみ)

機能:

プログラマブル電源の固有の識別コードを返します。

構文:

*IDN?

戻り値:

<string>には製造業者、型番、シリアル番号、およびフアームウエアバージョンが含まれます。

例:

*IDN?により、GW, PST-3202,0,,FW1.00 を返します。

*OPC

機能:

コマンド形式 (*OPC)では、保留中のオペレーションがすべて完了したとき標準イベントステータスレジスタ(SESR)のオペレーション完了ビット(ビット 0)をセットします。

クエリ形式(*OPC?)では、プログラマブル電源が保留中のオペレーションを完了したときに出力キューに ASCII コードの 1 を出力するようにプログラマブル電源に命じます。

構文:

*OPC

*OPC?

戻り値:

1

*RCL

機能:

以前に保存した設定データをメモリから呼び出します。

注意 PSS シリーズおよび PSH シリーズは、この機能を持っていません。

構文:

*RCL_<NR1>

<NR1>は、0～99 の範囲内です。

例:

*RCL_12 は、メモリアドレス 12 に保存した設定データを呼び出します。

*RST (クエリなし)

機能:

プログラマブル電源のすべての制御定をデフォルト値にしますが保存した設定は消去されません。デフォルト状態と等価なパネル設定を以下に示します。

前面パネルの操作部	デフォルト設定
OUTPUT	OFF

CURRENT SET	0
VOLTS SET	0
OCP SET	OFF
DELAY	1 sec
AUTO SET	OFF
RECALL(memory location)	00
OVP SET	最大値(OVP 設定についてはユーザーマニュアルを参照願います)
OUTPUT MODE (INDEP/ SERIES/PARALLEL)	INDEP
STEP SET	最大値(ユーザーマニュアルを参照願います)
RECALL RANGE	START 00 END 05 CYCLE 1

構文:

*RST

*SAV

機能:

設定データを特定のメモリアドレスに保存します(PSS シリーズオよび PSH シリーズはこの機能を持っていません)。

構文:

*SAV_<NR1>

<NR1>は、0 ～ 99 の範囲内です。

例:

*SAV 01 で現在の設定データをメモリのアドレス 1 に保存します。

*SRE

機能:

サービスリクエストイネーブルレジスタ(SRER)の内容を設定します。
クエリ形式では SRER の内容を返します。 SRER のビット 6 は常にゼロです。
SRER のビットは、SBR のビットに対応します。

構文:

*SRE <NR1>

*SRE?

戻り値:

<NR1>は、0～255 範囲内です。

例:

*SRE_7 は、SRER のビットを 00000011 に設定します。

*SRE? で応答が 3 のとき、 SRER は 0000 0011 に設定されています。

*STB? (クエリのみ)

機能:

(SBR) *STB? によるステータスバイトレジスタ(SBR) の問合わせを行うと、
ステータスレジスタ内で 1 にセットされたビットに対応する 10 進数を返します。

構文:

*STB?

戻り値:

<NR1>は、0～255 の範囲内の値です。

例:

*STB?は SBR が 2 進値 0101 0001 を含む場合には 81 を返します。

*TST? (クエリのみ)

機能:

セルフテストと RAM および ROM のテスト

構文:

*TST?

戻り値:

0 | -300

例:

*TST?により、試験に合格であれば 0 が返されます。

*TST?により、試験に不合格であれば-300 が返されます。

*WAI (クエリなし)

機能:

保留中のオペレーションが完了するまでデバイスがその先のコマンドあるいはクエリを実行しないようにさせます。

構文:

*WAI

:CHANnel<x>:CURRent

機能:

特定のチャネルの出力電流値の設定または問合わせを行います。

構文:

:CHANnel<x>:CURRent <NR2>

:CHANnel<x>:CURRent?

注意 <x>は 1、2、または 3 の値を取り得ます。 <NR2>については仕様を参照してください。

注意 <x>は、PSS シリーズおよび PSH シリーズでは取り得る値は 1 のみです。

戻り値:

<NR2>

例:

:CHANnel1:CURRent_2.0 により、チャネル 1 の電流制限を 2.0 A.

:CHANnel1:CURRent? により、チャネル 1 の電流制限値が 0.012 A に設定されている場合には 0.012 を返します。

:CHANnel<x>:VOLTage

機能:

特定のチャネルの出力電圧値の設定または問合せを行います。

構文:

:CHANnel<x>:VOLTage_<NR2>

:CHANnel<x>:VOLTage?

注意 <x>は 1、2、または 3 の値を取り得ます。 <NR2>については仕様を参照してください。

注意 <x>は、PSS シリーズおよび PSH シリーズでは取り得る値は 1 のみです。

戻り値:

<NR2>

例:

:CHANnel1:VOLTage_12.0 により、チャネル 1 の電圧制限を 12.0 V に設定します。

:CHANnel1:VOLTage?により、チャネル 1 の電圧制限値が 2.34 V に設定されている場合には 2.34 を返します。

:CHANnel<x>:MEASure:CURRent?(クエリのみ)

機能:

特定のチャネルの実際の出力電流を読み出します。

構文:

:CHANnel<x>:MEASure:CURRent?

注意 <x>は 1、2、または 3 の値を取り得ます。

注意 <x>は、PSS シリーズおよび PSH シリーズでは取り得る値は 1 のみです。

戻り値:

<NR2>

例:

:CHANnel1:MEASure:CURRent?により、負荷電流が 1.234A の場合には 1.234 を返します。

:CHANnel<x>:MEASure:VOLTage?(クエリのみ)

機能:

特定のチャネルの実際の出力電圧を返します。

構文:

:CHANnel<x>:MEASure:VOLTage?

注意 <x>は、1、2、または 3 の値を取り得ます。

注意 :<x>は、PSS シリーズおよび PSH シリーズでは取り得る値は 1 のみです。

戻り値:

<NR2>

例:

:CHANnel1:MEASure:VOLTage? により、チャネル 1 の出力電圧が 11.55 V の場合には 11.55 を返します。

:CHANnel<x>:PROTection:CURRent

機能:

特定のチャネルの過電流保護ステータスの設定あるいは問合わせを行います。

構文:

:CHANnel<x>:PROTection:CURRent_<Boolean>

:CHANnel<x>:PROTection:CURRent?

注意 <x>は 1、2、または 3 の値を取り得ます。

注意 <x>は、PSS シリーズおよび PSH シリーズでは取り得る値は 1 のみです。

戻り値:

0 | 1

例:

:CHANne1:PROTection:CURRent_0 により、過電流保護をオフに設定します。

過電流保護設定がオンの場合には、:CHANnel1:PROTection:CURRent?のコマンドにより 1 という値が返されます。

:CHANnel<x>:PROTection:VOLTage

機能:

特定のチャネルの過電圧保護値の設定あるいは問合わせを行いなす。

構文:

:CHANnel<x>:PROTection:VOLTage_<NR2>

:CHANnel<x>:PROTection:VOLTage?

注意 <x>は 1、2、または 3 の値を取り得ます。<NR2>については仕様を参照してください。

注意 <x>は、PSS シリーズおよび PSH シリーズでは取り得る値は 1 のみです。

戻り値:

<NR2>

例:

:CHANnel1:PROTection:VOLTage_12.0 により、チャネル 1 過電圧保護制限を 12.0 V に設定します。

:CHANnel1:PROTection:VOLTage?により、チャネル 1 過電流制限が 2.34 V に設定されている場合には 2.34 を返します。.

:OUTPut:COUPle:TRACking

機能:

チャネル 1 とチャネル2の出力を直列モード、並列モードあるいは独立動作モードに変更します。 (PSS シリーズではこの機能を持っていません)。

構文:

:OUTPut:COUPle:TRACking_<NR1>

:OUTPut:COUPle:TRACking?

<NR1>は、0(独立)、 1(並列、)または 2(直列)の値を取り得増す。

戻り値:

0 | 1 | 2

例:

:OUTPut:COUPle:TRACking_2 により、チャネル1およびチャネル 2 の出力を直列モードに設定します。

出力が並列モードの場合には、OUTPut:COUPle:TRACking? のコマンドによって 1 という値が返されます。

注意 PSS シリーズおよび PSH シリーズは、この機能を持っていません。

OUTPut:PROTection:CLEar (クエリ形式なし)

機能:

プログラマブル電源のパネル上の保護メッセージ(OTP, OVP, OCP)をすべてクリアします。

構文:

OUTPut:PROTection:CLEar

パネル面に保護メッセージが表示されている場合には、プログラマブル電源はそれ以降の設定を受け付けません。このコマンドを使って表示メッセージをクリアしてからそれ以降の設定を実行します。

例:

OUTPot:PROTection:CLEar

OUTPut:STATe

機能:

出力をオンまたはオフンに設定します。

構文:

OUTPut:STATe_<Boolean>

OUTPut:STATe?

<Boolean>は、0(オフ)または 1(オン)の値を取り得ます。

戻り値:

0 | 1

例:

OUTPut:STATe_1 により、電源をオンにします。

電源の出力が無効にされているときには、 OUTPut:STATe?によって 0 という値を返します。

STATus:OPERation:CONDition?(クエリのみ)

機能:

オペレーションレジスタの内容を返します。しかし、プログラマブル電源は状態通知のためにオペレーションレジスタを利用しません。

構文:

STATus:OPERation:CONDition?

戻り値:

<NR1>

例:

STATus:OPERation:CONDition? によって、0 という値を返します。

STATus:OPERation:ENABle

機能:

イネーブルマスクの設定あるいは問合わせを行います。イネーブルマスクによって、イペントレジスタ内のマスクされた状態がサマリービットとして報告されるようにします。イネーブルレジスタ内のビットは1(真)であり関連のイベントビットが1(真)に変化した場合には、関連のサマリービットは1(真)に変化します。これは

16 ビットのレジスタですが、15 ビット(ビット 0～14)のみを使っています。
ビット 15 は常に 0 です。

構文:

STATus:OPERation:ENABle_<NR1>

STATus:OPERation:ENABle?

<NR1> は、0～32767 の整数です。

戻り値:

<NR1>

例:

STATus:OPERation:ENABle_32767 により、レジスタの全 15 ビットを 1 にセットします。

STATus:OPERation:ENABle?によって 0 という値が返されるときには、レジスタの全 15 ビットはゼロです。

STATus:OPERation:EVENt(クエリのみ)

機能:

オペレーションレジスタの内容を返してクリアします。

構文:

STATus:OPERation:EVENt?

戻り値:

<NR1>

例:

STATus:OPERation:EVENt? により、0 という値を返します。

STATus:PRESet

機能:

オペレーションレジスタとクエスチョナブルレジスタをゼロに設定します。

構文:

STATus:PRESet

STATus:QUEStionable:CONDition? (クエリのみ)

機能:

クエスチョナブルレジスタの内容を返します。コンデイションレジスタの読み出し後、内容は保持されます。

構文:

STATus:QUEStionable:CONDition?

戻り値:

<NR1>

例:

STATus:QUEStionable:CONDition? により、0 という値を返します。

STATus:QUEStionable:ENABle

機能:

イネーブルマスクの設定あるいは問合わせを行います。イネーブルマスクによって、イペントレジスタ内のマスクされた状態がサマリビットとして報告されるようにします。イネーブルレジスタ内のビットは1(真)であり関連のイベントビットが1(真)に変化した場合には、関連のサマリビットは 1(真)に変化します。これは 16 ビットのレジスタですが、15 ビット(ビット 0～14)のみを使っています。ビット 15 は常に 0 です。

構文:

STATus:QUEStionable:ENABle_<NR1>

STATus:QUEStionable:ENABle?

<NR1>は、0～ 32767 の整数です。

戻り値:

<NR1>

例:

STATus:QUEStionable:ENABle_32767 により、レジスタの全 15 ビットを 1 にセットします。

STATus:QUEStionable:ENABle? によって 0 という値が返されたときには、レジスタの全 15 ビットは 0 です。

STATus:QUEStionable:EVENt(クエリのみ)

機能:

クエスチョナブルレジスタの内容を返してクリアします。応答はセットされたビットの 2 進値を要約した 10 進値です。

構文:

STATus:QUEStionable:EVENt?

戻り値:

<NR1>

例:

STATus:QUEStionable:EVENt?は、0 という値を返します。

SYSTem:AUTO:CYCLe

機能:

実行回数の設定あるいは問合わせを行います

注意　PSS シリーズおよび PSH シリーズは、この機能を持っていません。

構文:

SYSTem:AUTO:CYCLe_<NR1>

SYSTem:AUTO:CYCLe?

<NR1>は、0～ 99999 の範囲あるいは無限大です。

戻り値:

<NR1>

例:

SYSTem:AUTO:CYCLe_8 により、自動サイクルをオンにし、設定を 8 回繰り返します。

SYSTem:AUTO:CYCLe_0 により自動サイクルをオンにし、設定を無限に繰り返します。

If the command SYSTem:AUTO:CYCLe?コマンドが 0 を返す場合には、無限を意味します。

SYSTem:AUTO:DELay

機能:

現在応答しているメモリテータスの遅延時間を設定します。

注意　PSS シリーズおよび PSH シリーズは、この機能を持っていません。

構文:

SYSTem:AUTO:DELay_<NR1>

SYSTem:AUTO:DELay?

<NR1>は、1～59999 の範囲内で、単位は 100ms です。

戻り値:

<NR1>

例:

SYSTem:AUTO:DELay_1 は、メモリの特定部分について自動遅延時間を 100ms に設定します。

SYSTem:AUTO:DELay_1000 は、メモリの指定した部分の自動選択時間を 100 秒に設定します。以前の自動遅延が完了するまで次のメモリアドレスの自動遅延は実行されません。 SYSTem:AUTO:DELay? コマンドによって 5 という数値が返された場合には、LCD に表示される現在のメモリアドレスの遅延が 500ms であることを示します。

SYSTem:AUTO:END

機能:

連続自動実行の最終アドレスを設定します。

注意　PSS シリーズおよび PSH シリーズは、この機能を持っていません。

構文:

SYSTem:AUTO:END_<NR1>

SYSTem:AUTO:END?

<NR1>は、1～99 の範囲内であり、START の値以上でなければなりません。

戻り値:

<NR1>

例:

SYSTem:AUTO:END 8 は、現在のデバイスのメモリアドレス 8 を自動実行の最終アドレスに設定します。

SYSTem:AUTO:END? のコマンドが 99 という値を返す場合には、アドレス 99 が最終アドレスとして設定されていることを示します。

SYSTem:AUTO:STARt

機能:

連続自動実行の最終アドレスを設定します。

注意 PSS シリーズおよび PSH シリーズは、この機能を持っていません。

構文:

SYSTem:AUTO:STARt_<NR1>

SYSTem:AUTO:STARt?

<NR1>は、0～99 の範囲内であり、END の値以下でなければなりません。

戻り値:

<NR1>

例:

SYSTem:AUTO:STARt_0 は、現在のデバイスのメモリアドレス 0 を自動実行の開始アドレスに設定します。

SYSTem:ATUO:STARt?のコマンドが 2 という値を返す場合には、アドレス 2 が開始アドレスとして設定されていることを示します。

SYSTem:AUTO:STATe

機能:

自動実行設定の設定または問合わせを行います。

注意 PSS シリーズおよび PSH シリーズは、この機能を持っていません。

構文:

SYSTem:AUTO:STATe_<Boolean>

SYSTem:AUTO:STATe?

<Boolean>は、0(オフ)または 1(オン)の値を取り得ます。

戻り値:

0 | 1

例:

SYSTem:AUTO:STATe_1 により自動実行をオンにします。

SYSTem:ERRor? (クエリのみ)

機能:

エラー/イベントキューから次のエラーメッセージを取り出します。
問合わせの結果はエラー番号とエラーメッセージ本文です。

構文:

SYSTem:ERRor?

戻り値:

<string>

例:

SYSTem:ERRor? が 0 の場合は、「エラーなし」です。

SYSTem:MEMory? (クエリのみ)

機能:

パネルに表示されている現在のメモリアドレスを読み出します。

注意 PSS シリーズおよび PSH シリーズは、この機能を持っていません。

構文:

SYSTem:MEMory?

戻り値:

<NR1>

SYSTem:VERSion? (クエリのみ)

機能:

デバイスの SCPI バージョンを返します。

構文:

SYSTem:VERSion?

戻り値:

1994.0

7 ステータスとエラー報告

組になったステータスレジスタによって電源内部処理の状態を知ることができます。ステータスレジスタおよびステータスとイベントの報告の方法については、SCPI の推奨に従っています。

システムの構成

ステータスとイベントの報告方法の概要を図 7 に示します。図中の各構成要素は、システム内でのある種のイベントの読み出し、報告、およびイネーブルを行うことのできるいくつかのレジスタとキューを表します。

電源内の特定のイベントがステータスレジスタ内のビットをセットするときには、どのビットがセットされているかを読み取ることによってどの種類のイベントが発生したかを知ることができます。

ステータスレジスタの各ビットはイネーブルレジスタのビットに対応します。イベントがステータスバイトレジスタに報告されるためには、イネーブルビットが 1 でなければなりません。

サービスリクエスト(SRQ)は最も起こりにくいイベントです。SRQ は GP-IB に割込みをリクエストしてシステムコントローラにイベントを知らせます。

ステータスレジスタ

プログラマブル電源には 2 種類のステータスレジスタが組み込まれています。

オペレーションステータスレジスタ(コンデイション、イベント、およびイネーブル)

クエスチョナブルステータスレジスタ(コンデイション、イベント、およびイネーブル)

下位ノードであるクエスチョナブルとオペレーションのそれぞれには、コンデイション、イベント、およびイネーブルの 3 個の 16 ビットレジスタがあります。

これら 3 種類のレジスタと各レジスタに関連するコマンドの順序関係を図 8 に示します。

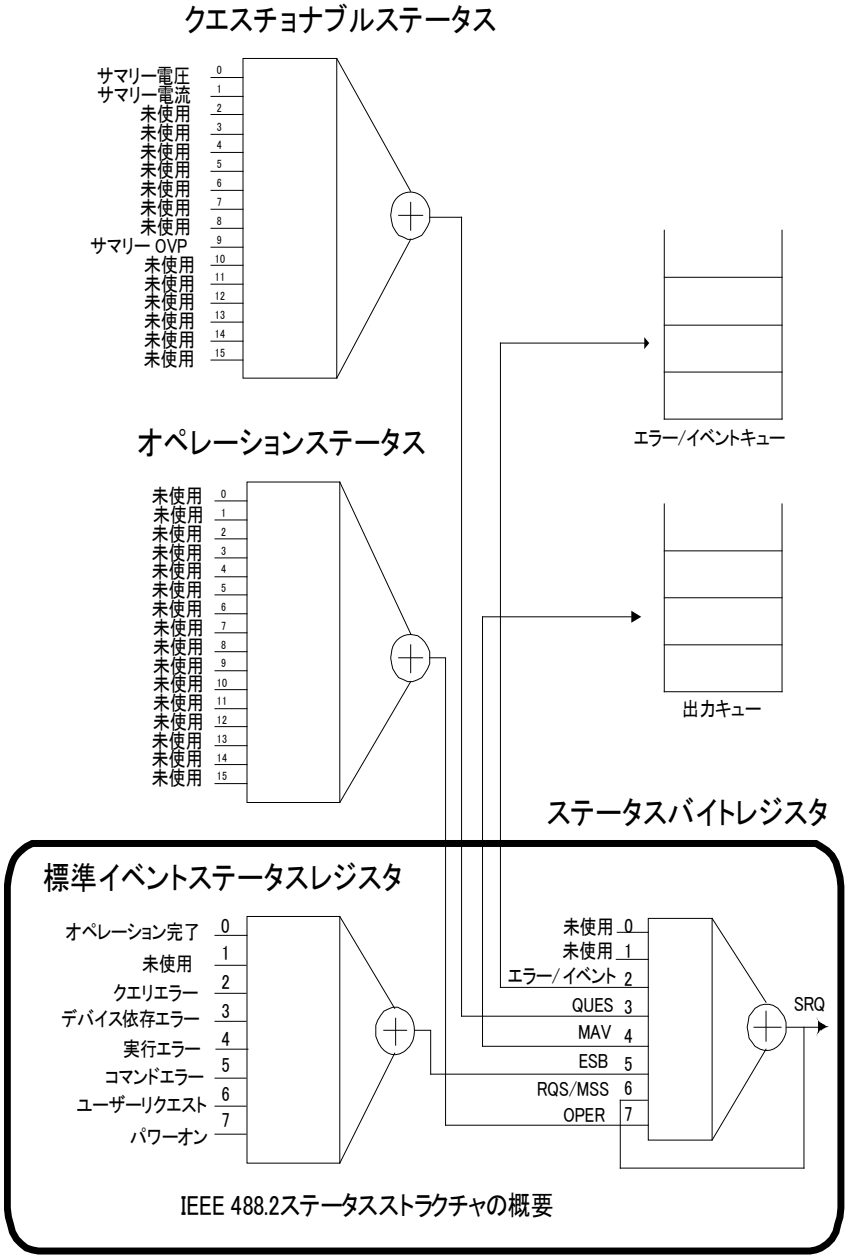

図7、ステータスレジスタと関連の連続との図式的関係

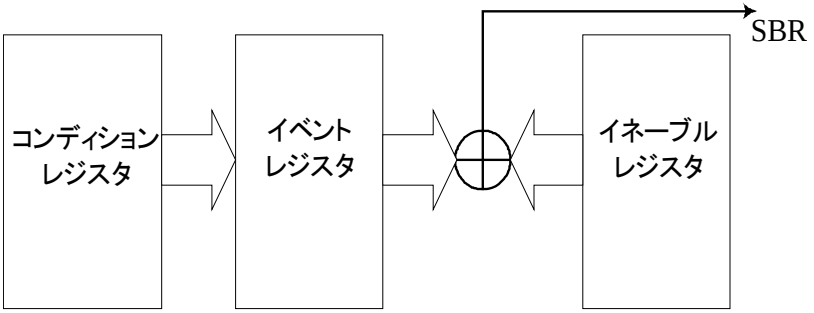

図 8:ステータスレジスタと関連コマンド

コンデイションレジスタは、読み出し専用のレジスタであり、計測器の現在の状態をモニタします。コンデイションレジタは、リアルタイムで更新され、入力のラッチあるいはバッファは行われません。コンデイションレジスタがモニタする状態が真になると、その状態のビットも真(1)になります。状態が偽のときにはビットは 0 です。

読み出し専用のイベントレジスタは、偽から真への状態変化があればラッチします。イベントレジスタのビットがセットされると、コンデイションレジスタの対応するビットが変化しても影響を受けません。ビットはコントローラが読み出すまでセットされたままです。

*CLS (クリアステータス)コマンドによってイベントレジスタをクリアします。

クエスチョナブルステータスレジスタ

16 ビットのクエスチョナブルステータスレジスタのビット割当てを表 4 に示します。

表 4 : クエスチョナブルステータスレジスタ

ビット 15	ビット 14	ビット 13	ビット 12	ビット 11	ビット 10	ビット 9	ビット 8
	*未使用	*未使用	*未使用	*未使用	*未使用	OVP	OCP

ビット 7	ビット*6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
未使用	未使用	未使用	OTP	未使用	未使用	CC	CV

STATus:QUEStionable:CONDtion? コマンドは、クエスチョナブルコンディションレジスタを読み出しますが、クリアはしません。

STATus:QUEStionable:EVENt?コマンドは、クエスチョナブルイベントステータスレジスタを読み出し、クリアします。

オペレーションステータスレジスタ

16 ビットのオペレーションステータスレジスタのビット割当てを表 5 に示します。

表 5 ： オペレーションステータスレジスタ

ビット 15	ビット 14	ビット 13	ビット 12	ビット 11	ビット 10	ビット 9	ビット 8
	*未使用	*未使用	*未使用	*未使用	*未使用	未使用	*未使用
ビット 7	ビット*6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
未使用	未使用	未使用	未使用	未使用	未使用	未使用	未使用

ステータスレジスタ

プログラマブル電源には IEEE-488.2 の規格で定義されている 2 つのステータスレジスタが含まれています。

ステータスバイトレジスタ (SBR)

標準イベントステータスレジスタ (SESR)

ステータスバイトレジスタ (SBR):SBR (表 6) は他のすべてのレジスタおよびキューのステータスを要約します。

表 6:ステータスバイトレジスタ (SBR)

ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
OPER	RQS/MSS	ESB	MAV	QUES	E/E	未使用	未使用

ビット 0　未使用です。このためビットは常に 0 です。

ビット 1　未使用です。このためビットは常に 0 です。

ビット 2　(エラーおよびイベント)はエラーイベントキュー内に読み出しを待っているエラーコードがあることを示します。

ビット 3　(QUES(クエスチョナブル))は QESR(クエスチョナブルイベントステータスレジスタ)の要約ビットです。このビットが 1 のときには、ステータスがイネーブルされ, QUES に存在することを示します。

ビット 4　(MAV (メッセージアベィラブル))は出力が出力キュー内で利用可能であることを示します。

ビット 5　(ESB(イベントステータスビット))は標準イベントステータスレジスタ(SESR)の要約ビットです。このビットが 1 のときには、ステータスがイネーブルされ SESR 内に存在することを示します。

ビット 6　(RQS,(リクエストサービス))はシリアルポールから得られ、プログラマブル電源が GP-IB コントローラからのサービスを要求していることを表示します。

ビット 7　(OPER(オペレーション))は OESR (オペレーションイベントステータスレジスタ))の要約ビットです。

SBR の内容を読み出すためにはシリアルポールあるいは*STB?クエリを使います。SBR 内のビットが標準イベントステータスレジスタ(SESR)および出力キューの内容に従って設定およびクリアされます。

標準イベントステータスレジスタ*(SESR)*: SESR を表 7 に示します。

表 7: 標準イベントステータスレジスタ (SESR):SESR を表 7 に示します。

ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0

PON	URQ	CME	EXE	DDE	QYE	未使用	OPC

ビット 0 (OPC(オペレーション完了))はオペレーションが完了したことを示します。このビットは*OPC コマンドの後、保留中のオペレーションがすべて完了したときにアクティブになります。

ビット1 常に 0 です。

ビット 2 (QYE(クエリエラー))はコマンドまたはクエリのプロトコルエラーを示します。

ビット 3 (DDE(デバイスエラー))はデバイスエラーが起こったことを示します。

ビット 4 (EXE(実行エラー))はプログラマブル電源がコマンドまたはクエリを実行しているときにエラーが発生したことを示します。

ビット 5 (CME(コマンドエラー))はプログラマブル電源がコマンドまたはエラーを解析しているときにエラーが発生したことを示します。

ビット 6 (USR(ユーザーリクエスト))は LOCAL ボタンが押されたことを示します。

ビット7 (PON(パワーオン))は電源がオンにされたことを示します。

SESR を読み出すためには *ESR?クエリを使用します。SESR を読み出して、新しいイベントの情報を入力できるようにレジスタのビットをクリアします。

イネーブルレジスタ

イネーブルレジスタは、特定のイベントをステータスバイトレジスタおよび SRQ に報告するかどうかを判定します。プログラマブル電源には以下のイネーブルレジスタがあります。

イベントステータスイネーブルレジスタ (ESER)

(オペレーションイネーブルレジスタ)

(クエスチョナブルイネーブルレジスタ)

サービスリクエストイネーブルレジスタ(SRER)

イネーブルレジスタのビットの 1 つが 1 であり、ステータスレジスタ内の対応するビットが 1 のときには、イネーブルレジスタ論理 AND 演算を行い、ステータスバイレジスタ内でセットするビットを制御する出力は 1 です。

各種のコマンドによりイネーブルレジスタ内のビットがセットされます。

イネーブルレジスタとセットするコマンドを以下に示します。

イベントステータスイネーブルレジスタ (ESER):

ESER はどの種類のイベントが SBR 内のイベントステータスビット(ESB)に要約されるかを制御します。ESER 内のビットは SESR 内のビットに対応します。

ESER のビットを設定するためには*ESE コマンドを使用します。 ESER を読み出すためには*ESE?クエリを使用します。

オペレーションイネーブルレジスタ:オペレーションイネーブルレジスタが本電源内にありますが、オペレーションレジスタは状態を報告しません。

クエスショナプルイネーブルレジス:

クエスショナプルイネーブルレジスタはどの種類のイベントが SBR 内の QUES ステータスビットに要約されるかを制御します。

クエスチョナプナプルイネーブルレジスタ内のビットを設定するためには STATus:QUEStionable:ENABle コマンドを使用します。

読み出すためには STATus:QUEStionable:ENABle?クエリを使用します。

サービスリクエストイネーブルレジスタ (SRER):

SRER は SBR 内のどのビットがサービスリクエストを生成するかを制御します。
SRER を設定するためには*SRE コマンドを使用します。
SRER を読み出すためには*SRE?クエリを使用します。

キュー

出力キューがプログラマブル電源に組み込まれています。

出力キュー:プログラマブル電源はクエリに対する応答を IEEE-488.2 プログラマブル電源がメッセージターミネータの後で新しいコマンドまたはクエリのメッセージを受け取ると、プログラマブル電源はその都度このキューをクリアしてリセットします。以前のクエリに対する応答が失われないようにするために、パソコンがクエリの応答を読み出した後で新しいコマンド(またはクエリ)を送信しなければなりません。

エラー/イベントキュー

エラーまたはイベントが発生すると、出力キューがメッセージを保存します。

出力キューはメッセージを FIFO(ファーストインファーストアウト)方式で保存して報告します。

SYSTem:ERRor?クエリにより出力キューから次の項目を読み出します。

出力キューがオーバフローすると、エラーメッセージが-350(キューオーバフロー)になり、キューが読み出されてクリアされるまで次のメッセージの保存と報告を行うことができません。

エラーメッセージ

エラーメッセージプログラマブル電源の SCPI エラーメッセージを表 8 に示します。

表 8 プログラマブル電源のエラーメッセージ

SCPI のエラーコードと説明	SESR のビット
0, (エラーなし)	
-100, (コマンドエラー)	5
-200, (実行エラー)	4
-221, (設定の矛盾)	4
-221, (設定の矛盾、タイマー設定エラー)	4
-221, (設定の矛盾、過電圧保護設定エラー)	4
-221, (設定の矛盾、電圧設定エラー)	4
-221, (設定の矛盾、電流設定エラー)	4
-221, (設定の矛盾、呼び出し設定エラー)	4
-221, (設定の矛盾、設定保存エラー)	4
-222, (範囲外データ)	4
-222, (範囲外データ、電圧が大きすぎます)	4
-222, (範囲外データ、電流が大きすぎます)	4
-222, (範囲外データ、電圧が小さすぎます)	4
-222, (範囲外データ、電流が小さすぎます)	4

-240, （ハードウエアエラー）	4
-300, （デバイス固有エラー）	3
-300, （デバイス固有エラー、校正電流エラー）	3
-300, （デバイス固有エラー、校正電圧エラー）	3
-300, （デバイス固有エラー、過電圧保護エラー）	3
-310, （システムエラー）	3
-313, （構成メモリ喪失）	3
-330, （セルフテスト不合格）	3
-330, （セルフテスト不合格、CPU テストエラー）	3
-330, （セルフテスト不合格、RAM テストエラー）	3
-330, （セルフテスト不合格、ROM テストエラー）	3
-330, （セルフテスト不合格、 DAC/ADC テストエラー）	3
-350, （キューのオーバフロー）	3
-410, （クエリが割込みを受けた）	2
-420, （クエリが未定）	2
-430, （クエリのデッドロック発生）	2

お問い合わせ

製品についてのご質問等につきましては、下記まで
お問い合わせください。
TEL:03-5823-5656 FAX:03-5823-5655

E-Mail:info@instek.co.jp
HomePage:http://www.instek.co.jp

株式会社 インステック ジャパン
〒101-0032 東京都千代田区岩本町 1-3-3